IN THE CLAIMS:

Please cancel claims 67 – 166 without prejudice or disclaimer as to the subject matter therein.

B3

167. (New) A computer-implemented method for programmatically creating a graphical program, comprising:

creating a first program, wherein the first program is executable to programmatically create the new graphical program;

executing the first program, wherein said executing comprises programmatically creating the new graphical program, wherein said programmatically creating the new graphical program comprises:

creating a plurality of graphical program objects in the new graphical program; and

interconnecting the plurality of graphical program objects in the new graphical program;

wherein the interconnected plurality of graphical program objects comprise at least a portion of the new graphical program.

168. (New) The method of claim 167, wherein said programmatically creating the new graphical program creates the new graphical program without any user input specifying the plurality of graphical program objects or the interconnection of the plurality of graphical program objects program during said creating.

169. (New) The method of claim 167, wherein the new graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the new graphical program.

170. (New) The method of claim 167, wherein the new graphical program comprises a diagram portion comprising a plurality of interconnected nodes and a user interface portion;

wherein said programmatically creating the new graphical program comprises creating the diagram portion and the user interface portion.

171. (New) The method of claim 167, wherein the new graphical program comprises a data flow diagram.

172. (New) The method of claim 167, wherein the new graphical program is a virtual instrument.

173. (New) The method of claim 167, wherein said executing the first program occurs in a first computing environment;

wherein said first computing environment is connected to a second computing environment;

wherein said executing the first program comprises sending information from the first computing environment to the second computing environment;

wherein the new graphical program is created in the second computing environment.

174. (New) The method of claim 167,

wherein the first program specifies creation of the new graphical program.

175. (New) The method of claim 167, further comprising:

receiving information from a user;

wherein said programmatically creating comprises programmatically creating the new graphical program at least partially based on the information received from the user.

176. (New) The method of claim 167,

wherein the first program comprises a client program that calls an application programming interface (API) to programmatically create the new graphical program;

wherein said executing comprises the first program calling the API to perform said creating and said interconnecting.

177. (New) The method of claim 167,

wherein the first program comprises a server program

wherein said executing comprises the first program receiving calls from a client program.

178. (New) The method of claim 167, wherein the new graphical program comprises a block diagram, wherein the plurality of graphical program objects comprise at least one function node placed in the block diagram.

B3

179. The method of claim 167, wherein the new graphical program includes a block diagram, wherein the plurality of graphical program objects comprise at least one programmatic structure placed in the block diagram.

180. (New) The method of claim 167, wherein the plurality of graphical program objects comprise at least one graphical loop structure.

181. (New) The method of claim 167, wherein the plurality of graphical program objects comprise at least one graphical case structure.

182. (New) The method of claim 167, wherein said interconnecting the plurality of graphical program objects comprises displaying a connection between an input of a first graphical program object and an output of a second graphical program object.

183. (New) The method of claim 167, wherein said programmatically creating the new graphical program comprises:

creating one or more user interface objects, wherein the one or more user interface objects perform one or more of providing input to or displaying output from the new graphical program.

184. (New) The method of claim 167, wherein the new graphical program includes a user interface panel, wherein plurality of graphical program objects comprise at least one user interface object placed in the user interface panel.

185. (New) The method of claim 184, wherein the user interface object comprises at least one of:

a user interface input object placed in the user interface panel for providing user input to the new graphical program; and/or

a user interface output object placed in the user interface panel for viewing output of the new graphical program.

186. (New) The method of claim 167, wherein said programmatically creating the new graphical program further comprises:

obtaining a reference to a graphical program object, wherein the reference is used to manipulate the graphical program object.

187. (New) The method of claim 167, wherein said programmatically creating the new graphical program further comprises:

performing at least one of getting or setting a property of a graphical program object.

188. (New) The method of claim 167, wherein said programmatically creating the new graphical program further comprises:

invoking a method on a graphical program object.

189. (New) The method of claim 167, wherein the first program is a first graphical program.

190. (New) The method of claim 189, wherein the first graphical program includes at least one object creation node for programmatically creating at least one graphical program object in the new graphical program;

wherein said programmatically creating the new graphical program comprises including the at least one graphical program object in the new graphical program.

191. (New) The method of claim 190, wherein the first graphical program further includes a property node;

wherein said programmatically creating the new graphical program comprises the property node getting or setting a property of the graphical program object.

192. (New) The method of claim 191, wherein the object creation node outputs a reference to the graphical program object;

wherein the property node receives as input the reference to the graphical program object;

wherein the property node gets or sets a property of the graphical program object specified by the reference to the graphical program object.

193. (New) The method of claim 190, wherein the first graphical program further includes an invoke node;

wherein said programmatically creating the new graphical program comprises the invoke node invoking a method on the graphical program object.

194. (New) The method of claim 193, wherein the object creation node outputs a reference to the graphical program object;

wherein the invoke node receives as input the reference to the graphical program object;

wherein the invoke node invokes a method on the graphical program object specified by the reference to the graphical program object.

195. (New) The method of claim 193, wherein the invoked method connects the graphical program object to another graphical program object in the new graphical program.

196. (New) The method of claim 195, wherein said connecting the graphical program object to said another graphical program object comprises connecting an input of the graphical program object to an output of said another graphical program object.

197. (New) The method of claim 193, wherein the invoked method performs one of 1) moving the graphical program object to another location in the new graphical program; and 2) resizing the graphical program object in the new graphical program.

198. (New) The method of claim 190, wherein said creating the first graphical program comprises:

displaying the object creation node;

specifying a graphical program object class for the object creation node;

wherein the at least one graphical program object included in the new graphical program is of the graphical program object class.

199. (New) The method of claim 198, wherein said creating the first graphical program further comprises:

specifying a graphical program object sub-class for the object creation node;

wherein the graphical program object included in the new graphical program is of the object sub-class.

200. (New) The method of claim 190, wherein said creating the first graphical program comprises:

displaying the object creation node;

specifying position information to the object creation node;

wherein the at least one graphical program object included in the new graphical program is positioned in the new graphical program at a location based on the position information.

201. (New) The method of claim 190, wherein said creating the first graphical program comprises:

displaying the object creation node;

specifying owner reference information for the object creation node, wherein the owner reference information designates an owner entity;

wherein the at least one graphical program object is included in the new graphical program as a member of the owner entity.

202. (New) The method of claim 201, wherein the owner entity is an entity from the group consisting of: 1) the new graphical program and 2) another graphical program object of the new graphical program.

203. (New) The method of claim 189, wherein the first graphical program includes a graphical program creation node for programmatically creating the new graphical program.

204. (New) The method of claim 203, wherein said creating the first graphical program comprises:

displaying the graphical program creation node;

specifying a new graphical program type for the graphical program creation node;

wherein said creating the new graphical program comprises creating the new graphical program of the specified new graphical program type.

205. (New) The method of claim 204, wherein the graphical program creation node includes a type input;

wherein said specifying a new graphical program type for the graphical program creation node comprises connecting type information to the type input of the graphical program creation node.

206. (New) The method of claim 203, wherein said creating the first graphical program comprises:

displaying the graphical program creation node;

specifying a template graphical program for the graphical program creation node;

wherein said creating the new graphical program comprises creating the new graphical program based on the template graphical program.

207. (New) The method of claim 206, wherein the graphical program creation node includes a template input;

wherein said specifying a template graphical program for the graphical program creation node comprises connecting information specifying an existing graphical program to the template input of the graphical program creation node.

208. (New) The method of claim 203, wherein said creating the first graphical program comprises:

displaying the graphical program creation node;

specifying a reference to a server program for the graphical program creation node;

wherein said creating the new graphical program comprises the server program creating the new graphical program.

209. (New) The method of claim 208, wherein the server program is an application instance of a graphical programming development environment.

210. (New) The method of claim 208, wherein the graphical program creation node includes a server program reference input;

wherein said specifying a reference to a server program for the graphical program creation node comprises connecting information specifying a server program to the server program reference input of the graphical program creation node.

211. (New) The method of claim 208, wherein said executing the first graphical program is performed in a first computing system;

wherein said server program executes in a second computing system;

wherein the first computing system is connected to the second computing system.

212. (New) The method of claim 189, wherein said creating the first graphical program comprises:

displaying a graphical program creation node, wherein the graphical program creation node is operable to programmatically create the new graphical program;

displaying an object creation node, wherein the object creation node is operable to programmatically create at least one graphical program object in the new graphical program; and

configuring the object creation node with one or more inputs.

213. (New) The method of claim 212, further comprising:

connecting the graphical program creation node to the object creation node;

wherein the graphical program creation node outputs a reference to the new graphical program;

wherein said connecting the graphical program creation node to the object creation node comprises connecting the reference to the new graphical program to an input of the object creation node.

214. (New) The method of claim 212, further comprising configuring the graphical program creation node with one or more inputs, wherein said configuring the graphical program creation node with one or more inputs comprises performing one or more of:

1) specifying a new graphical program type for the graphical program creation node; 2) specifying a template graphical program for the graphical program creation node; and 3) specifying a server reference for the graphical program creation node.

215. (New) The method of claim 214, wherein a server reference is specified for the graphical program creation node;

wherein said executing the first graphical program comprises executing program instructions on a first computer;

wherein the server reference references a server program running on a second computer;

wherein the second computer is connected to the first computer via a network;

wherein said creating the new graphical program in response to said executing the first graphical program comprises the server program creating the new graphical program.

216. (New) The method of claim 212, wherein said configuring the object creation node with one or more inputs comprises performing one or more of:

1) specifying an object class for the object creation node; 2) specifying an object sub-class for the object creation node; 3) specifying position information to the object creation node; and 4) specifying owner reference information for the object creation node.

217. (New) The method of claim 189, wherein the first graphical program includes a plurality of object creation nodes each for programmatically creating a graphical program object in the new graphical program, wherein said plurality of object creation nodes includes a first object creation node for creating a first graphical program object in the new graphical program and includes a second object creation node for creating a second graphical program object in the new graphical program;

wherein said executing the first graphical program comprises including the first graphical program object and the second graphical program object in the new graphical program;

wherein the first graphical program further includes a node operable to connect the first graphical program object to the second graphical program object;

wherein said executing the first graphical program includes connecting the first graphical program object to the second graphical program object.

218. (New) The method of claim 189,

wherein the first graphical program includes a graphical program creation node for programmatically creating the new graphical program;

wherein the first graphical program includes at least one object creation node for programmatically creating at least one graphical program object in the new graphical program;

wherein said executing the first graphical program includes:

executing the graphical program creation node, wherein said executing the graphical program creation node causes creation of the new graphical program; and

executing the object creation node, wherein said executing the object creation node causes inclusion of the graphical program object in the new graphical program.

219. (New) A memory medium, comprising:

a first program, wherein the first program is operable to programmatically create a new graphical program;

wherein the first program comprises program instructions that are operable to:

create a plurality of graphical program objects in the new graphical program; and

interconnect the plurality of graphical program objects in the new graphical program;

wherein the interconnected plurality of graphical program objects comprise at least a portion of the new graphical program.

220. (New) The memory medium of claim 219, wherein the first program is operable to programmatically create the new graphical program without any user input

specifying selection of graphical program objects and interconnection of graphical program objects.

221. (New) The memory medium of claim 219, wherein the new graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the new graphical program.

222. (New) The memory medium of claim 219, wherein the new graphical program comprises a diagram portion comprising a plurality of interconnected nodes and a user interface portion;

wherein the first program is operable to programmatically create the diagram portion and the user interface portion.

223. (New) The memory medium of claim 219, wherein the new graphical program comprises a data flow diagram.

224. (New) The memory medium of claim 219, wherein the new graphical program is a virtual instrument.

225. (New) The memory medium of claim 219, wherein the first program is operable to execute in a first computing environment;

wherein said first computing environment is connected to a second computing environment;

wherein during execution the first program is operable to send information from the first computing environment to the second computing environment;

wherein the new graphical program is created in the second computing environment.

226. (New) The memory medium of claim 219

wherein the first program specifies creation of the new graphical program.

227. (New) The memory medium of claim 219,

wherein the first program is operable to receive information from a user and programmatically create the new graphical program at least partially based on the information received from the user.

228. (New) The memory medium of claim 219,

wherein the first program comprises a client program that calls an application programming interface (API) to programmatically create the new graphical program;

wherein during execution the first program is operable to call the API to perform said creating and said interconnecting.

229. (New) The memory medium of claim 219,

wherein the first program comprises a server program;

wherein during execution the first program is operable to receive calls from a client program.

230. (New) The memory medium of claim 219, wherein the new graphical program comprises a block diagram, wherein the plurality of graphical program objects comprise at least one function node placed in the block diagram.

231. (New) The memory medium of claim 219, wherein the new graphical program includes a block diagram, wherein the plurality of graphical program objects comprise at least one programmatic structure placed in the block diagram.

232. (New) The memory medium of claim 219, wherein the plurality of graphical program objects comprise at least one graphical loop structure.

233. (New) The memory medium of claim 219, wherein the plurality of graphical program objects comprise at least one graphical case structure.

234. (New) The memory medium of claim 219, wherein, in interconnecting the plurality of graphical program objects, the program instructions are operable to display a connection between an input of a first graphical program object and an output of a second graphical program object.

235. (New) The memory medium of claim 219, wherein the program instructions are further operable to:

create one or more user interface objects, wherein the one or more user interface objects perform one or more of providing input to or displaying output from the new graphical program.

236. (New) The memory medium of claim 219, wherein the new graphical program includes a user interface panel, wherein plurality of graphical program objects comprise at least one user interface object placed in the user interface panel.

237. (New) The memory medium of claim 236, wherein the user interface object comprises at least one of:

a user interface input object placed in the user interface panel for providing user input to the new graphical program; and/or

a user interface output object placed in the user interface panel for viewing output of the new graphical program.

238. (New) The memory medium of claim 219, wherein the program instructions are further operable to:

obtain a reference to a graphical program object, wherein the reference is used to manipulate the graphical program object.

239. (New) The memory medium of claim 219, wherein the program instructions are further operable to:

perform at least one of getting or setting a property of a graphical program object.

240. (New) The memory medium of claim 219, wherein the program instructions are further operable to:

invoke a method on a graphical program object.

241. (New) A memory medium, comprising:

a first graphical program, wherein the first graphical program includes a graphical program creation node for programmatically creating a new graphical program, wherein the first graphical program also includes a first object creation node for creating a first object in the new graphical program and includes a second object creation node for creating a second object in the new graphical program;

wherein the first graphical program is executable to:

create the new graphical program;

include the first object and the second object in the new graphical program; and

connect the first object to the second object.

242. (New) A memory medium, comprising:

a first program, wherein the first program is operable to programmatically modify an existing graphical program;

wherein the first program comprises program instructions that are operable to:

create at least one graphical program object in the existing graphical program; and

interconnect the at least one graphical program object to an existing graphical program object in the existing graphical program.

243. (New) The memory medium of claim 242, wherein the program instructions are operable to modify the existing graphical program without any user input specifying the at least one graphical program object or interconnection of the at least one graphical program object to the existing graphical program.

244. (New) The memory medium of claim 242, wherein the existing graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the existing graphical program.

245. (New) The memory medium of claim 242, wherein the existing graphical program includes a plurality of graphical program objects;

wherein the program instructions are further operable to modify one or more graphical program objects in the existing graphical program.

246. (New) The memory medium of claim 242, wherein the existing graphical program includes a plurality of user interface objects;

wherein the program instructions are further operable to modify one or more user interface objects in the existing graphical program.

247. (New) The memory medium of claim 242, wherein the existing graphical program includes a user interface portion;

wherein the program instructions are further operable to modify the user interface portion.

248. (New) The memory medium of claim 242, wherein the existing graphical program is a virtual instrument.

249. (New) The memory medium of claim 242, wherein the first program executes in a first computing environment;

wherein the first computing environment is connected to a second computing environment;

wherein the existing graphical program is stored in the second computing environment;

wherein during execution the first program is operable to send information from the first computing environment to the second computing environment;

wherein the existing graphical program is modified in the second computing environment.

250. (New) The memory medium of claim 242, wherein the first program is a first graphical program.

251. (New) The memory medium of claim 242, wherein the first graphical program includes at least one modify node;

wherein the modify node is executable to modify one or more graphical program objects in the existing graphical program.

252. (New) The memory medium of claim 251, wherein the modify node comprises one of an invoke node or a property node;

wherein the property node is operable to get or set a property of the existing graphical program or a property of an object of the existing graphical program;

wherein the invoke node is operable to invoke a method on the existing graphical program or on an object of the existing graphical program.

253. (New) The memory medium of claim 252, wherein the first graphical program includes a property node;

wherein a reference to the existing graphical program is provided to the property node;

wherein the property node is configured to get or set a property of the existing graphical program.

254. (New) The memory medium of claim 252, wherein the first graphical program includes an invoke node;

wherein a reference to the existing graphical program is provided to the invoke node;

wherein the invoke node is configured to invoke a particular method on the existing graphical program.

255. (New) The memory medium of claim 241, wherein the first graphical program also includes a node for obtaining a graphical program object reference;

wherein the reference to the existing graphical program is provided to the node for obtaining a graphical program object reference;

wherein the node for obtaining a graphical program object reference is configured to obtain a reference to a particular object of the existing graphical program;

wherein the first graphical program includes a modify node;

wherein the reference to the particular object of the existing graphical program is provided to the modify node;

wherein the modify node is configured to modify the particular object of the existing graphical program.

256. (New) A computer-implemented method for programmatically creating a graphical program, comprising:

creating a first program, wherein the first program includes a graphical program creation function for programmatically creating a new graphical program, wherein the first program also includes an object creation function for programmatically including an object in the new graphical program;

executing the first program;

creating the new graphical program in response to said executing the first program; and

including the object in the new graphical program in response to said executing the first program.

257. (New) The method of claim 256, wherein the first program is a graphical program;

wherein the graphical program creation function comprises a graphical program creation node;

wherein the object creation function comprises an object creation node.

258. (New) The method of claim 256, wherein the first program is a text-based program.

259. (New) The method of claim 258, wherein the graphical program creation function comprises a method call to create the new graphical program;

wherein the object creation function comprises a method call to create the object.

260. (New) The method of claim 258, wherein the text-based program obtains a reference to a software component;

wherein the software component enables the text-based program to perform the method call to create the new graphical program;

wherein the software component enables the text-based program to perform the method call to create the object.

261. (New) The method of claim 260, wherein the software component interfaces with a server program;

wherein the server program receives the method call to create the new graphical program;

wherein the server program creates the new graphical program;

wherein the server program receives the method call to create the object;

wherein the server program creates the object.

262. (New) The method of claim 260, wherein the software component is an ActiveX component.

263. (New) A computer-implemented method for programmatically accessing a graphical program, comprising:

creating a first program, wherein the first program includes a function for obtaining a reference to an existing graphical program, wherein the first program also includes a modify function, wherein the modify function is operable to add a graphical program object to the existing graphical program;

executing the first program; and

adding a graphical program object in the existing graphical program in response to said executing the first program.

264. (New) A system for programmatically creating a graphical program, comprising:

a computer system including a CPU and memory;

wherein the memory stores a first program, wherein the first program specifies creation of a new graphical program, wherein the first program is executable to programmatically create the new graphical program;

wherein the CPU is operable to execute the first program to programmatically create the new graphical program, wherein, in executing the first program, the CPU is operable to:

create a plurality of graphical program objects in the new graphical program; and

interconnect the plurality of graphical program objects in the new graphical program;

wherein the interconnected plurality of graphical program objects comprise at least a portion of the new graphical program.

265. (New) The system of claim 264, wherein, in response to said CPU executing the first program, the first program is operable to interface with a server program;

wherein the server program is operable to programmatically create the new graphical program in response to said interfacing.

266. (New) The system of claim 265, wherein the server program is an application instance of a graphical programming development environment.

267. (New) The system of claim 265, wherein said computer system is a first computer system, the system further comprising:

a second computer system;

wherein the server program executes in the second computer system;

wherein the first computer system is connected to the second computer system via a network.

268. (New) The system of claim 264,

wherein at least one of the plurality of graphical program objects comprises a structure node.

269. (New) The system of claim 264, wherein the first program is a first graphical program.

270. (New) The system of claim 269, wherein the first graphical program includes at least one object creation node for programmatically creating at least one graphical program object in the new graphical program;

wherein said creating the new graphical program comprises including the at least one graphical program object in the new graphical program in response to said executing the first graphical program.

271. (New) The system of claim 269, wherein the first graphical program includes a graphical program creation node for programmatically creating the new graphical program.

272. (New) A system for programmatically creating or accessing a graphical program, comprising:

a computer system including a CPU and memory;

a client program executing in the computer system, wherein the client program performs API calls to programmatically create or access a graphical program;

a server program operable to receive the client program calls to programmatically create or access a graphical program and operable to perform the respective operations.

273. (New) The system of claim 272, wherein the server program executes on another computer system, wherein said another computer system is connected to said computer system via a network.

274. (New) The system of claim 272, wherein the client program performs said calls to programmatically create or access a graphical program by obtaining a reference to a software component and invoking methods of the software component;

wherein the software component is operable to perform the operations of programmatically creating or accessing the graphical program.

275. (New) The system of claim 272, wherein the client program performs said calls to programmatically create or access a graphical program by obtaining a reference to a software component and invoking methods of the software component;

wherein the software component relays the client program calls to the server program.

276. (New) The system of claim 272, wherein the server program is a graphical programming environment application.

277. (New) The system of claim 272, wherein the client program is a client graphical program;

wherein the client graphical program includes a graphical program creation node for programmatically creating a new graphical program;

wherein the client graphical program also includes an object creation node for programmatically creating a graphical program object in the new graphical program;

wherein said API calls to programmatically create or access a graphical program comprise calls resulting from executing the graphical program creation node and the object creation node.

278. (New) The system of claim 277, wherein the client graphical program further includes a property node for getting or setting a property of the graphical program object.

279. (New) The system of claim 277, wherein the client graphical program further includes an invoke node for invoking a method on the graphical program object.

280. (New) The system of claim 279, wherein the object creation node is a first object creation node for programmatically creating a first graphical program object in the new graphical program;

wherein the graphical program also includes a second object creation node for programmatically creating a second graphical program object in the new graphical program;

wherein the invoked method connects the first graphical program object to the second graphical program object.

281. (New) A memory medium comprising a client program for programmatically creating a new graphical program, wherein the client program comprises:

a means for instantiating the new graphical program;

a means for adding an object to the new graphical program;

a means for getting or setting properties of the new graphical program or the object;

a means for invoking methods on the new graphical program or the object.

282. (New) The memory medium of claim 281, wherein the client program is a graphical program;

wherein said means for instantiating the new graphical program comprises a graphical program creation node;

wherein said means for adding an object to the new graphical program comprises an object creation node;

wherein said means for getting or setting properties of the new graphical program or the object comprises a property node;

wherein said means for invoking methods on the new graphical program or the object comprises an invoke node.

283. (New) A memory medium comprising a client program for programmatically accessing a graphical program, wherein the client program comprises:

a means for obtaining a reference to the graphical program;

a means for obtaining a reference to a particular object of the graphical program;

a means for getting or setting properties of the graphical program or the object;

a means for invoking methods on the graphical program or the object.

284. (New) A computer-implemented method for programmatically creating a graphical program, comprising:

creating a first graphical program, wherein the first graphical program includes a graphical program creation node for programmatically creating a new graphical program, wherein the first graphical program also includes at least one object creation node for programmatically creating at least one graphical program object in the new graphical program;

B3

executing the first graphical program;

creating the new graphical program in response to said executing the first graphical program; and

including the at least one graphical program object in the new graphical program in response to said executing the first graphical program.